

REMARKS

Claims 1, 3-12, 14-23, and 25-35 are pending. These same claims stand rejected. Applicants have added new dependent claims 36 and 37. Claims 2, 6, 13, 17, 24, 26-28, and 33 have been canceled.

Applicants would like to thank the Examiner for conducting an interview on May 13, 2008 with Applicants' representatives Peter Trahms-Neudorfer and Han Gim. The contents of the interview are identified in the relevant sections below.

Claim Rejections - 35 USC § 102

Claims 1, 3, 4, 7-15, 18-23, 25, 29-31, 34, and 35 have been rejected under 35 U.S.C. § 102 as being anticipated by anonymous publication "Protected Error Handling For Microprocessor-Based Systems" (hereinafter Anonymous). These same claims have also been rejected under 35 U.S.C. § 102 as being anticipated by U.S. Publication Number 2004/0133777 to Kiriansky et al. (hereinafter Kiriansky). It is respectfully submitted that claims 1, 3, 4, 7-15, 18-23, 25, 29-31, 34, and 35 are patentable for the reasons set forth below.

Applicants have amended claim 1 to recite:

A method of generating and storing a list of safe exception handlers, comprising:

- identifying valid exception handlers;
- including a safe exception section for compiled objects, said safe exception section comprising addresses of valid exception handlers in the section;
- marking the compiled object if the compiled object references or contains at least one valid exception handler, said marking operable to identify that the object is associated with a valid exception handler;
- generating at least one list of valid exception handlers, said at least one list comprising the valid exception handlers included or referenced in the compiled objects and
- storing said at least one list of valid exception handlers in a protected area during program execution.

As Applicants explained during the May 13 interview, “anonymous” generally describes placing exception handlers in read only memory to prevent corruption during program execution. The exceptions handlers in read only memory can also include provisions for accessing exception handlers in writable portions of memory after performing validity checks. The passages cited from Kiriansky et al. generally describe executing an instruction if the instruction is from a page that was unmodified during program execution. An example provided is to execute the instruction if the instruction is from a page that is protected in read only memory.

The above references do not disclose including a safe exception section for compiled objects comprising addresses of valid exception handlers in the section, marking the compiled object if the compiled object references or contains at least one valid exception handler, and generating at least one list of valid exception handlers comprising the valid exception handlers included or referenced in the compiled objects, as now recited in claim 1. Both references rely upon protecting the actual exception handler code, rather than including a safe exception section for each object and marking the object as safe such that, for example, an operating system can determine whether the executable supports safe exception handling.

Similar amendments have been made to Claim 12.

Claim 23 has been amended to recite:

A system for executing safe exceptions, comprising:
a processor configured to receive an event; and
an exception dispatcher system configured to determine an exception handler for the event, further comprising determining if the exception handler is valid by comparing the exception handler to a list of valid exception handlers and determining if the exception handler is unaltered, and otherwise determine that the exception handler is invalid, wherein said list of valid exception handlers are located in a protected area during program execution and said list comprises exception handlers included or referenced in a safe exception section for compiled executable objects.

The above references do not disclose a list of valid exception handlers comprising exception handlers included or referenced in a safe exception section for compiled executable

objects, as now recited in claim 23. Both references rely upon protecting the actual exception handler code, rather than providing a safe exception section for each executable object and marking the object as safe such that, for example, an operating system can determine whether the executable supports safe exception handling.

Therefore, it is respectfully submitted that claims 1, 12, and 23 are patentable, and their dependent claims including claims 3, 4, 7-11, 13-15, 18-22, 25, 29-31, 34, and 35 are patentable for at least the same reasons. During the May 13 interview, the Examiner agreed that the above amendments appear to overcome the present rejection. Withdrawal of the rejection of claims 1, 3, 4, 7-15, 18-23, 25, 29-31, 34, and 35 under 35 U.S.C. § 102 is respectfully requested.

Claim Rejections - 35 USC § 103

Claims 1, 3, 4, 7-15, 18-23, 25, 29-31, 34, and 35 have been rejected under 35 U.S.C. § 103 as being unpatentable over U.S. Pub. No. 2004/0064712 (Arthur et al.) in view of U.S. Pat. No. 7,243,340 (Tobin). It is respectfully submitted that claims 1, 3, 4, 7-15, 18-23, 25, 29-31, 34, and 35 are patentable for the reasons set forth below.

Arthur et al. is directed to preventing the debugging of software during execution in order to prevent reverse engineering of the code (see Abstract). To prevent debugging, the exception handlers are replaced by substitute exception handlers (see page 2 paragraph 0019). The substitute exception handlers are patched into the O/S (page 4 paragraph 0053) during execution of the program (page 3 paragraph 0051). Arthur et al. does not disclose nor render obvious including a safe exception section for compiled objects comprising addresses of valid exception handlers in the section, marking the compiled object if the compiled object references or contains at least one valid exception handler, and generating at least one list of valid exception handlers comprising the valid exception handlers included or referenced in the compiled objects, as now recited in claim 1. Similar amendments have been made to Claim 12, which recite similar features to those described in claim 1. Furthermore, Arthur et al. does not disclose nor render obvious a list of valid exception handlers comprising exception handlers included or referenced in a safe exception section for compiled executable objects, as now recited in claim 23.

Therefore, claims 1, 12, and 23, are patentable, and their dependent claims including claims 3, 4, 7-11, 13-15, 18-22, 25, 29-31, 34, and 35 are patentable for at least the same reasons.

Withdrawal of the rejection of claims 1, 3, 4, 7-15, 18-23, 25, 29-31, 34, and 35 under 35 U.S.C. § 103 is respectfully requested.

Claims 5, 16, 26, 27, and 32 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Arthur et al in view of Tobin as applied to claims 1, 23 or 26, and in further view of U.S. Pat. No. 5,628,016 (Kukol). Claims 5, 16, 26, 27, and 32 variously depend from independent claims 1, 12, and 23, and are therefore patentable for at least the reasons set forth above. Kukol fails to cure the deficiencies of Arthur et al. Kukol generally describes exception handling, but does not teach or suggest including a safe exception section for compiled objects comprising addresses of valid exception handlers in the section, marking the compiled object if the compiled object references or contains at least one valid exception handler, and generating at least one list of valid exception handlers comprising the valid exception handlers included or referenced in the compiled objects, nor an exception dispatcher system configured to determine if an exception handler is valid by comparing the exception handler to a list of valid exception handlers and determining if the exception handler is unaltered, and otherwise determine that the exception handler is invalid, wherein said list comprises exception handlers included or referenced in a safe exception section for compiled executable objects. For at least the foregoing reasons, reconsideration of the rejection of claims 5, 16, 26, 27, and 32 under 35 U.S.C. § 103(a) is respectfully requested.

Claims 6, 17, and 28 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Arthur et al. in view of Tobin and in further view of U.S. Publication No. 20020169999 (Bhansali). Claims 6, 17, and 28 depend from independent claims 1, 12, and 23 respectively, and are therefore patentable for at least the reasons set forth above with respect to these claims. Bhansali fails to cure the deficiencies of Arthur et al. Bhansali generally describes exception handlers, but does not teach or suggest including a safe exception section for compiled objects comprising addresses of valid exception handlers in the section, marking the compiled object if the compiled object references or contains at least one valid exception handler, and generating at

least one list of valid exception handlers comprising the valid exception handlers included or referenced in the compiled objects, nor an exception dispatcher system configured to determine if an exception handler is valid by comparing the exception handler to a list of valid exception handlers and determining if the exception handler is unaltered, and otherwise determine that the exception handler is invalid, wherein said list comprises exception handlers included or referenced in a safe exception section for compiled executable objects. Therefore, reconsideration of the rejection of claims 6, 17, and 28 under 35 U.S.C. § 103(a) is respectfully requested.

Claim 33 has been rejected under 35 U.S.C. § 103(a) as being unpatentable over Arthur et al. in view of Tobin, and in further view of Kukol and Bhansali. Claim 33 depends from independent claim 23, and is therefore patentable for at least the reasons set forth above with respect to claim 23. The combination of Kukol and Bhansali fails to cure the deficiencies of Arthur et al. Kukol and Bhansali generally describe exception handlers, but do not teach or suggest including a safe exception section for compiled objects comprising addresses of valid exception handlers in the section, marking the compiled object if the compiled object references or contains at least one valid exception handler, and generating at least one list of valid exception handlers comprising the valid exception handlers included or referenced in the compiled objects, nor an exception dispatcher system configured to determine if an exception handler is valid by comparing the exception handler to a list of valid exception handlers and determining if the exception handler is unaltered, and otherwise determine that the exception handler is invalid, wherein said list comprises exception handlers included or referenced in a safe exception section for compiled executable objects. Therefore, reconsideration of the rejection of claim 33 under 35 U.S.C. § 103(a) is respectfully requested.

DOCKET NO.: MSFT-1650/302481.01
Application No.: 10/602,952
Office Action Dated: March 27, 2008

PATENT
REPLY FILED UNDER EXPEDITED
PROCEDURE PURSUANT TO
37 CFR § 1.116

CONCLUSION

In view of the foregoing amendments and remarks, Applicants submit that the above-identified application is in condition for allowance. Early notification to this effect is respectfully requested.

Date: May 27, 2008

/Michael J. Swope/
Michael J. Swope
Registration No. 38,041

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439